

Informe Laboratorio 3

Sección 1

Tobías Guerrero Cheuquepán
e-mail: tobias.guerrero_c@mail.udp.cl

Octubre de 2023

Índice

1. Descripción de actividades	2
2. Desarrollo (Paso 1)	3
2.1. Identificar en qué se destaca la red del informante del resto	4
2.2. Explica matemáticamente porqué se requieren más de 5000 paquetes para obtener la pass	4
2.3. Obtiene la password con ataque por defecto de aircrack-ng	5
2.4. Indica el tiempo que demoró en obtener la password	5
2.5. Descifra el contenido capturado	6
2.6. Describe como obtiene la url de donde descargar el archivo	6
3. Desarrollo (Paso 2)	8
3.1. Indica script para modificar diccionario original	8
3.2. Cantidad de passwords finales que contiene rockyou_mod.dic	9
4. Desarrollo (Paso 3)	10
4.1. Obtiene contraseña con hashcat con potfile	10
4.2. Identifica nomenclatura del output	11
4.3. Obtiene contraseña con hashcat sin potfile	12
4.4. Identifica nomenclatura del output	13
4.5. Obtiene contraseña con aircrack-ng	13
4.6. Identifica y modifica parámetros solicitados por pycrack	14
4.7. Obtiene contraseña con PyCrack	19

1. Descripción de actividades

Su informante quiere entregarle la contraseña de acceso a una red, pero desconfía de todo medio para entregársela (aún no llega al capítulo del curso en donde aprende a comunicar una password sin que nadie más la pueda interceptar). Por lo tanto, le entregará un archivo que contiene un desafío de autenticación, que al analizarlo, usted podrá obtener la contraseña que lo permite resolver. Como nadie puede ver a su informante (es informante y debe mantener el anonimato), él se comunicará con usted a través de la redes inalámbricas y de una forma que solo usted, como experto en informática y telecomunicaciones, logrará esclarecer.

1. Identifique cual es la red inalámbrica que está utilizando su informante para enviarle información. Obtenga la contraseña de esa red utilizando el ataque por defecto de aircrack-ng, indicando el tiempo requerido para esto. Descifre el contenido transmitido sobre ella y descargue de Internet el archivo que su informante le ha comunicado a través de los paquetes que usted ha descifrado.
2. Descargue el diccionario de RockyouLinks to an external site. (utilizado ampliamente en el mundo del pentesting). Haga un script que para cada string contenido en el diccionario, reemplace la primera letra por su letra en capital y agregue un cero al final de la password.
3. Todos los strings que comiencen con número toca eliminarlos del diccionario. Indique la cantidad de contraseñas que contiene el diccionario modificado debe llamarse rock-you_mod.dic A continuación un ejemplo de cómo se modifican las 10 primeras líneas del diccionario original.

2. Desarrollo (Paso 1)

Para iniciar el proceso de ataque, es esencial contar con aircrack-ng debidamente instalado. Además, es necesario identificar la tarjeta de red y configurarla en el modo monitor. La instalación de aircrack-ng se lleva a cabo mediante el siguiente comando:

```
1 sudo apt-get install aircrack-ng
```

Listing 1: Comando para instalar aircrack-ng

Una vez instalado, se utiliza el siguiente comando para identificar la interfaz, que en este caso corresponderá a “wlp2s0”.

```
1 iwconfig
```

Listing 2: Comando para ver las interfaces de red

Una vez que se conocen los parámetros necesarios, procederemos a poner la tarjeta en modo monitor con la ayuda de los siguientes comandos:

```
1 sudo service NetworkManager stop
2 sudo airmon-ng check
3 sudo airmon-ng check kill
4 sudo systemctl stop avahi-daemon
5 sudo airmon-ng start wlp2s0
```

Listing 3: Comandos para configurar el modo monitor

Una vez que todo esté configurado, utilizaremos el siguiente comando para escanear el entorno y descubrir las redes presentes, lo cual nos proporcionará la información necesaria para la siguiente fase:

```
1 sudo airodump-ng wlp2s0
```

Listing 4: Comandos para escanear aire

De aquí se obtendrá lo siguiente:

Kill them using 'airmon-ng check kill' before putting
CH 6][Elapsed: 6 s][2023-10-17 05:44

BSSID	PNR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
80:1F:8C:E0:E8:84	-86	1	0	0	11	130	WPA3	CCMP	OWE <length: 0>
98:FC:11:86:B6:B9	0	6	96	8	6	130	WPA2	CCMP	PSK Telematica
F6:4B:29:1C:85:38	-98	6	0	0	11	1			<length: 0>
B0:48:7A:D2:DD:74	-44	9	102	0	6	54e	WEP	WEP	WEP
54:FB:78:35:DF:AF	-49	9	0	0	1	360	WPA2	CCMP	PSK AndroidNeto
58:EF:68:47:59:C8	-64	10	0	0	6	130	OPN		cableadaTelematica-Invitado
58:EF:68:47:59:C8	-64	6	0	0	6	130	WPA2	CCMP	PSK cableadaTelematica
80:1F:8C:E2:14:A7	-70	7	0	0	11	130	WPA2	CCMP	MGT Administrativos-UDP
84:1C:30:B5:EA:07	-69	11	0	0	10	130	WPA2	CCMP	PSK ZTE_B5EA07
80:1F:8C:E2:14:A3	-69	9	0	0	11	130	OPN		Alumnos-UDP
80:1F:8C:E1:B2:03	-83	4	0	0	1	130	OPN		Alumnos-UDP
80:1F:8C:E2:14:A1	-73	7	0	0	11	130	OPN		Invitados-UDP
80:1F:8C:E1:B2:07	-83	5	0	0	1	130	WPA2	CCMP	Administrativos-UDP
80:1F:8C:E1:B2:06	-82	3	0	0	1	130	WPA3	CCMP	OWE <length: 0>
80:1F:8C:E1:B2:05	-86	6	0	0	1	130	OPN		VIP-UDP
80:1F:8C:E1:B2:02	-77	5	0	0	1	130	WPA3	CCMP	OWE <length: 0>
80:1F:8C:E2:14:A4	-70	7	0	0	11	130	WPA3	CCMP	OWE <length: 0>
80:1F:8C:E2:14:A0	-66	2	0	0	11	130	WPA3	CCMP	SAE Sala Hibrida-UDP
84:08:1B:C6:B3:E9	-75	2	0	0	2	195	WPA2	CCMP	PSK FAMILIAGL_EXT
CC:04:A1:07:81:00	-75	6	0	0	13	130	WPA2	CCMP	PSK HUAWEI-B2368-D781DD
CC:ED:DC:1C:0E:71	-71	3	0	0	13	130	WPA2	CCMP	PSK JPablo
80:1F:8C:E2:14:A5	-70	7	0	0	11	130	OPN		VIP-UDP
80:1F:8C:E1:B2:01	-88	4	0	0	1	130	OPN		Invitados-UDP
18:35:01:80:60:81	-77	1	0	0	1	130	WPA2	CCMP	PSK VTR-1506154
C0:05:C2:E3:09:41	-77	2	0	0	11	130	WPA2	CCMP	PSK CAFM
80:1F:8C:E2:14:A6	-77	8	0	0	11	130	WPA3	CCMP	OWE <length: 0>
80:1F:8C:E2:14:A2	-69	13	0	0	11	130	WPA3	CCMP	OWE <length: 0>
E4:AB:89:67:33:90	-78	1	0	0	1	130	WPA2	CCMP	PSK Otakus depa
80:1F:8C:E1:B2:04	-78	5	0	0	1	130	WPA3	CCMP	OWE <length: 0>
18:6E:3E:0E:74:A0	-78	2	1	0	1	270	WPA2	CCMP	PSK LSA
48:D3:43:86:61:19	-87	2	0	0	1	130	WPA2	CCMP	PSK VTR-6709284
14:CC:20:E8:E8:35	-83	3	0	0	13	270	WPA2	CCMP	PSK JPablo_EXT
AC:7B:CC:1D:60:68	-84	4	0	0	1	130	WPA2	CCMP	PSK VTR-8492879
9C:9D:7E:22:19:90	-85	2	0	0	13	130	WPA2	CCMP	PSK Kata rep
18:35:01:90:C7:99	-71	2	0	0	1	130	WPA2	CCMP	PSK VTR-6733269
18:35:01:48:E8:39	-85	1	0	0	1	130	WPA2	CCMP	PSK VTR-5376275
00:94:4C:95:1B:A6	-80	1	0	0	10	130	WPA2	CCMP	PSK HUAWEI-B032-3BA6
26:96:82:26:A7:5E	-79	3	0	0	11	130	WPA2	CCMP	PSK Martin 2,4g
80:1F:8C:E1:B2:00	-71	3	0	0	1	130	WPA3	CCMP	SAE Sala Hibrida-UDP

Figura 1: BSSID obtenidas

2.1 Identificar en qué se destaca la red del informante del resto DESARROLLO (PASO 1)

A partir de este punto, podremos identificar la red mediante dos campos esenciales, uno correspondiente al **encode** y otro al **cipher**, ambos relacionados con WEP. Además, se obtendrá el BSSID necesario para capturar el tráfico, que en este caso es “B0:48:7A:D2:DD:74”.

2.1. Identificar en qué se destaca la red del informante del resto

En el mundo de las redes, encontramos diversos tipos de cifrado que desempeñan un papel fundamental en la protección de la privacidad y la seguridad de las comunicaciones. Algunos ejemplos de estos protocolos de cifrado incluyen WEP, WPA, WPA2 y WPA3. La red utilizada por nuestro informante se destaca del resto debido a su uso de WEP, un protocolo que, sin embargo, ha quedado obsoleto y se considera inseguro. Este protocolo presenta debilidades conocidas que lo hacen vulnerable a ataques.

2.2. Explica matemáticamente porqué se requieren más de 5000 paquetes para obtener la pass

Para entender por qué se necesitan al menos 5000 paquetes para descifrar una contraseña WEP, se utiliza el concepto de la “paradoja del cumpleaños”. Esta paradoja se fundamenta en conceptos de probabilidad y estadísticas, y nos proporciona una comprensión de por qué se requiere una cantidad sustancial de paquetes con el mismo valor de inicialización (IV) repetido. La fórmula relacionada con esta paradoja es la siguiente:

$$P(N) = 1 - e^{-\frac{n \cdot (n-1)}{2 \cdot N}} \quad (1)$$

En donde:

- $P(N)$ es la probabilidad de que ocurra una colisión en los valores IV.
- n es el número de paquetes recopilados.
- N es el número total de valores IV posibles.

Dentro del contexto de una red WEP, el valor NN se asocia con la longitud del IV, la cual en este caso consta de 24 bits, generando aproximadamente 16.7 millones de posibles IV. Para lograr una probabilidad significativamente alta de éxito en un ataque de fuerza bruta, se hace necesario acumular un número sustancial de paquetes, que debe superar los 5000, todos con el mismo IV repetido, a fin de que $P(N)$ sea lo suficientemente elevado, al menos al 50 %. Esto se debe a que, al incrementar el valor de n , la probabilidad de colisión aumenta proporcionalmente, lo que a su vez incrementa las perspectivas de adivinar con éxito la clave de cifrado.

En mi caso, se capturaron aproximadamente 73000 frames. Al sustituir este número en el cálculo, la probabilidad indicaría un 100 % de lograr encontrar la clave.

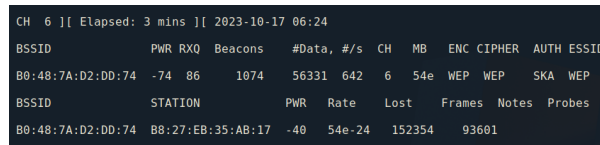
2.3. Obtiene la password con ataque por defecto de aircrack-ng

Después de completar toda la configuración mencionada anteriormente, procederemos a utilizar el comando para capturar mensajes. En este paso, especificaremos la dirección MAC de la red y el canal por el cual se está transmitiendo.

```
1 sudo airodump-ng -c 6 --bssid B0:48:7A:D2:DD:74 -w captura:paquetes wlp2s0
```

Listing 5: Comando para capturar mensajes

De aquí se tendrá:



BSSID	PWR	RXQ	Beacons	#Data	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
B0:48:7A:D2:DD:74	-74	86	1074	56331	642	6	54e	WEP	WEP	SKA	WEP
BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes				
B0:48:7A:D2:DD:74	B8:27:EB:35:AB:17	-40	54e-24	152354	93601						

Figura 2: Capturando mensajes

La captura se llevó a cabo durante un par de minutos y se guardó bajo el nombre “captura:paquetes”. Esta captura se utilizará para obtener la contraseña, para lo cual se empleará el siguiente comando:

```
1 sudo aircrack-ng -b B0:48:7A:D2:DD:74 captura:paquetes-01.cap
```

Listing 6: Comando para obtener la contraseña

Una vez que se ejecuta el comando, la terminal mostrará el valor de la contraseña. Esto se puede observar a continuación:

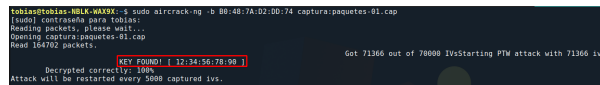


Figura 3: Contraseña obtenida

2.4. Indica el tiempo que demoró en obtener la password

Para calcular el tiempo que aircrack-ng tarda en obtener la contraseña, se empleará el mismo comando previamente mencionado. La diferencia radica en la adición del parámetro “time”, el cual proporciona el tiempo real que toma la ejecución del comando para obtener la contraseña. A continuación, se presenta el comando y los resultados:

```
1 time sudo aircrack-ng -b B0:48:7A:D2:DD:74 captura:paquetes-01.cap
```

Listing 7: Comando para obtener el tiempo de ejecución

```
tobias@tobias-NBLK-WAX9X: $ time sudo aircrack-ng -b 88:48:7A:02:DD:74 captura:paquetes-01.ca
p
[sudo] contraseña para tobias:
Reading packets, please wait...
Opening captura:paquetes-01.cap
Read 164702 packets. Got 71366 out of 70000 IVsStarting PTW attack with 71366 ivs.
KEY FOUND! [ 12:34:56:78:90 ]
Decrypted correctly: 100%
Attack will be restarted every 5000 captured ivs.

real    0m2.337s
user    0m0.058s
sys     0m0.041s
```

Figura 4: Tiempo que tarda en obtener la contraseña

Se puede observar que el proceso lleva aproximadamente 2.337 segundos para obtener la contraseña.

2.5. Descifra el contenido capturado

Para descifrar el contenido de la captura, se ejecutará el siguiente comando, el cual requerirá tanto la contraseña del paso anterior como la captura. A continuación, se presenta el comando:

```
1 sudo airdecap-ng -w 12:34:56:78:90 captura:paquetes-01.cap
```

Listing 8: Comando para descifrar el contenido

Una vez ejecutado se obtendrá:

```
tobias@tobias-NBLK-WAX9X:~$ sudo airdecap-ng -w 12:34:56:78:90 captura:paquetes-01.cap
Total number of stations seen      6
Total number of packets read      164702
Total number of WEP data packets   71445
Total number of WPA data packets   0
Number of plaintext data packets   0
Number of decrypted WEP packets    71445
Number of corrupted WEP packets    0
Number of decrypted WPA packets    0
Number of bad TKIP (WPA) packets   0
Number of bad CCMP (WPA) packets   0
```

Figura 5: Proceso de descifrado

Este comando generará un archivo DEC, correspondiente a “captura:paquetes-01-dec.cap”, el cual se podrá abrir con la ayuda del programa Wireshark, para posteriormente analizar los paquetes ICMP correspondientes.

2.6. Describe como obtiene la url de donde descargar el archivo

La captura descifrada obtenida en el paso anterior contiene varios paquetes ICMP generados mediante un ping por el punto de acceso (AP). A continuación, se presentan los paquetes:

2.6 Describe como obtiene la url de donde descargar el archivo DESARROLLO (PASO 1)

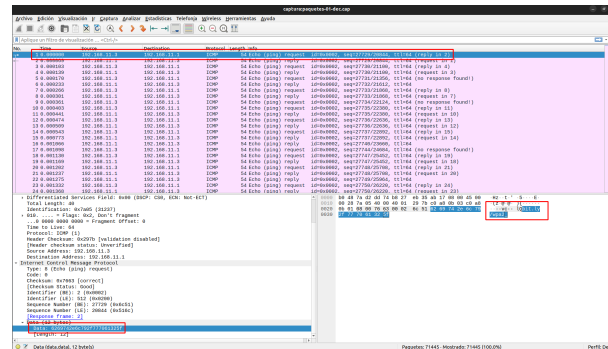


Figura 6: Paquetes ICMP con la URL obtenida

Se selecciona un paquete ICMP y se analiza su payload respectivo, donde se puede observar que se encuentra una URL en texto plano, la cual corresponde a `bit.ly/wpa2_`. Esta URL contiene otra captura, la cual se puede observar a continuación:

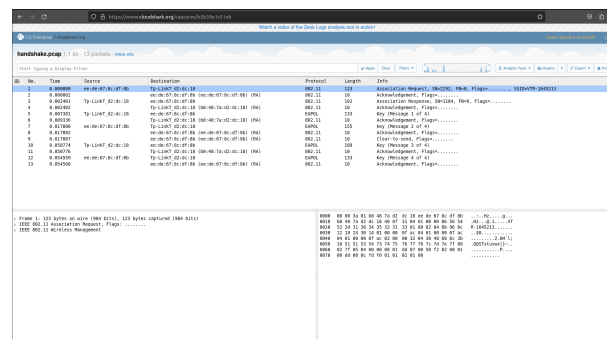


Figura 7: Captura contenida en la URL

3. Desarrollo (Paso 2)

3.1. Indica script para modificar diccionario original

Se solicita descargar el diccionario “rockyou.txt”, el cual será modificado con la ayuda de un script para cumplir con los requisitos previamente mencionados. Estos requisitos incluyen reemplazar la primera letra de cada cadena en el diccionario por su letra en mayúscula y agregar un 0 al final de la contraseña. Además, se deben eliminar todas las cadenas que comiencen con un número. A continuación, se presenta el código en Python que lleva a cabo estas modificaciones:

```

1 # Definir una funcion para modificar el diccionario
2 def modificar_diccionario(diccionario_file):
3
4 # Lista para almacenar las contraseñas modificadas
5     contraseñas_modificadas = []
6
7 # Abrir el archivo de diccionario original en modo lectura con el codec '
  latin-1'
8     with open(diccionario_file, 'r', encoding='latin-1') as archivo:
9         # Leer líneas del archivo
10            líneas = archivo.readlines()
11
12        # Recorrer cada línea del archivo
13        for línea in líneas:
14            # Eliminar espacios en blanco al principio y al final de la
  línea
15                línea = línea.strip()
16
17            # Verificar si la línea no está vacía y si no comienza con un
  número
18            if línea and not línea[0].isdigit():
19                # Modificar la primera letra a mayúscula y agregar un '0'
  al final
20                contraseña_modificada = línea[0].upper() + línea[1:] + '0'
21            ,
22
23            # Agregar la contraseña modificada a la lista
24            contraseñas_modificadas.append(contraseña_modificada)
25
26        # Guardar las contraseñas modificadas en un nuevo archivo
27        with open('rockyou_mod.dic', 'w') as archivo_modificado:
28            for contraseña_a in contraseñas_modificadas:
29                archivo_modificado.write(contraseña_a + '\n')
30
31        # Devolver la cantidad de contraseñas en el archivo modificado
32        return len(contraseñas_modificadas)
33
34 # Nombre del archivo de diccionario original
35 diccionario_original = 'rockyou.txt'

```


3.2 Cantidad de passwords finales que contiene rockyou_mod.dic DESARROLLO (PASO 2)

```
36 # Llamar a la funcion para modificar el diccionario
37 cantidad_contrase as_modificadas = modificar_diccionario(
    diccionario_original)
38
39 # Imprimir la cantidad de contraseñas en el archivo modificado
40 print(f'Se han modificado y guardado {cantidad_contrase as_modificadas}
    contraseñas en el archivo rockyou_mod.dic.')
```

Listing 9: Script que modifica el diccionario rockyou

3.2. Cantidad de passwords finales que contiene rockyou_mod.dic

Después de realizar todas las modificaciones necesarias, se obtiene un total de 11,059,725 contraseñas. A continuación, se muestra la cantidad y el aspecto del nuevo diccionario:

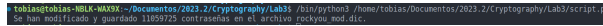


Figura 8: Cantidad de contraseñas obtenidas

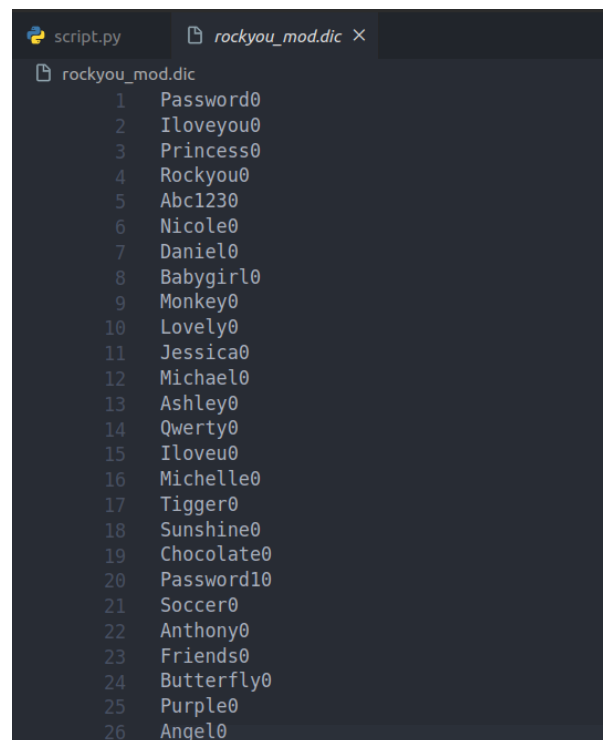


Figura 9: Nuevo diccionario

4. Desarrollo (Paso 3)

4.1. Obtiene contraseña con hashcat con potfile

Para obtener la contraseña, el primer paso consiste en instalar hashcat. Esto se llevará a cabo mediante los siguientes comandos:

```
1 sudo apt update
2 sudo apt upgrade
3 sudo apt install hashcat
```

Listing 10: Comando para instalar hashcat

Luego, para utilizar hashcat, el siguiente paso es convertir la captura descargada desde la página, es decir, “handshake.pcap”, a un archivo .hccapx. Este procedimiento se llevará a cabo utilizando la siguiente URL:<https://hashcat.net/cap2hashcat/>.

Handshake extraction successful: [Download](#)

```
hccapngtool 6.3.1 reading from 190198_1697598951.cap...
summary capture file
.....
file name.....: 190198_1697598951.cap
version (pcap/cap).....: 2.4 (very basic format without any additional information)
timestamp minimum (GPT).....: 12.10.2022 23:29:57
timestamp maximum (GPT).....: 12.10.2022 23:29:58
used capture interfaces.....: 1
link layer header type.....: DLT_IEEE802_11 (805) very basic format without any additional information about the quality
endianness (capture system).....: little endian
packets inside.....: 13
ESSID (total unique).....: 1
ASSOCIATIONREQUEST (total).....: 1
ASSOCIATIONREQUEST (PSK).....: 1
EAPOL messages (total).....: 4
EAPOL RSN messages.....: 4
EAPOLTIME gap (measured maximum msec).....: 33
EAPOL ANOUNCE error corrections (INC).....: not detected
EAPOL M1 messages (total).....: 1
EAPOL M2 messages (total).....: 1
EAPOL M3 messages (total).....: 1
EAPOL M4 messages (total).....: 1
EAPOL M4 messages (zeroed NONCE).....: 1
EAPOL pairs (total).....: 2
EAPOL pairs (best).....: 1
EAPOL pairs written to 22000 hash file...: 1 (RC checked)
EAPOL PSK2C (authorized).....: 1

Information: limited dump file format detected!
This file format is a very basic format to save captured network data.
It is recommended to use PCAP Next Generation dump file format (or pcapng for short) instead.
The PCAP Next Generation dump file format is an attempt to overcome the limitations
of the currently widely used (but very limited) libpcap (cap, pcap) format.
https://www.wireshark.org/docs/ugug_html_chunked/AppFiles.html#AppFilesCaptureFilesSection
https://github.com/pcapng/pcapng

Information: radiotap header is missing!
Radiotap is a de facto standard for 802.11 frame injection and
reception. The radiotap header format is a mechanism to supply
additional information about frames, from the driver to userspace
applications.
https://www.radiotap.org/

Information: missing frames!
This dump file does not contain undirected probe request frames.
An undirected probe request may contain information about the PSK.
It always happens if the capture file was cleaned or
it could happen if filter options are used during capturing.
That makes it hard to recover the PSK.

Information: missing frames!
This dump file does not contain enough EAPOL M1 frames.
It always happens if the capture file was cleaned or
it could happen if filter options are used during capturing.
That makes it impossible to calculate nonce-error-correction values.
```

Figura 10: Conversión archivo a .hccapx

Después de completar el proceso anterior, se obtendrá la contraseña utilizando el siguiente comando. En este caso, se utilizará el potfile, que es un archivo utilizado por herramientas para recuperar contraseñas. Hashcat guardará las contraseñas descifradas en el archivo 'potfile.txt'. A continuación, el comando:

```
1 hashcat -m 22000 190198_1697598951.hc22000 rockyou_mod.dic --potfile-path
   potfile.txt --force
```

Listing 11: Comando para obtener contraseña con hashcat con potfile

Una vez que se ejecute el comando, se obtendrá el siguiente resultado:

```

tobias@tobias-MILK-WAY3X:~/Documentos/2023_2/Cryptography/Lab3$ hashcat -m 22000 190198 1097508951.hc22000 rockyou_mod.dic --potfile-path potfile.txt --force
hashcat (v6.2.5) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.

OpenCL API (OpenCL 2.0 pocl 1.0 Linux, None+Asserts, RELOC, LLVM 11.1.0, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
* Device #1: pthread-AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx, 2539/5143 MB (1024 MB allocatable), BMCU

Minimum password length supported by kernel: 8
Maximum password length supported by kernel: 63

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Slow-Hash-SIMD-LOOP

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 2 MB

Dictionary cache built:
* Filename: rockyou_mod.dic
* Passwords: 11059725
* Bytes: 120186275
* Keyspace: 11059707
* Runtime: 1 sec

1013ac9576741b446d4339f9b96bf98-b0487ad2dc18-eede678cdf8b-VTR-1645213:Security0
Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PWKID+EAOL)
Hash.Target.....: 190198 1097508951.hc22000
Time.Started.....: Wed Oct 18 08:39:20 2023, (1 sec)
Time.Estimated...: Wed Oct 18 08:39:21 2023, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_mod.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 1603 K/s (8.54ms) @ Accel:64 Loops:256 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 2907/11059707 (0.03%)
Rejected.....: 1371/2907 (47.16%)
Restore.Point...: 1965/11059707 (0.02%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Magandako08 -> Dangerous0
Hardware.Mon.#1... Temp: 77C Util: 91%
Started: Wed Oct 18 08:37:49 2023
Stopped: Wed Oct 18 08:39:23 2023

```

Figura 11: Hashcat con potfile

De la figura anterior, se puede observar la contraseña obtenida, la cual es “Security0”.

4.2. Identifica nomenclatura del output

A continuación se analiza cada campo del output:

- **Device:** Indica el dispositivo que se está utilizando para realizar el ataque.
- **Password Length:** Indica el largo mínimo y máximo del formato compatible de las contraseñas que recibe hash.
- **Hashes:** Indica el número de hashes, de hashes únicos y de los salts únicos que se utilizan.
- **Rules:** Indica la cantidad de reglas utilizadas para el ataque.
- **Password length:** Indica la cantidad de optimizadores que se ocuparon el ataque.
- **Watchdog:** Indica el límite de temperatura, si la temperatura del dispositivo es de 90 grados Celcius el ataque se detendrá.
- **Host memory required:** Indica la cantidad de memoria necesaria para el ataque.
- **Dictionary cache built:** Indica detalles acerca del diccionario utilizado, como cantidad de contraseñas y el tamaño en bytes.

- **Hash descifrado:** Indica el hash descifrado
- **Session:** Indica información general de la sesión
- **Speed:** Velocidad con la que se descifra la información relacionada al rendimiento del ataque.
- **Started:** Inicio de ataque
- **Stopped:** Fin de ataque

4.3. Obtiene contraseña con hashcat sin potfile

Ahora, para obtener la contraseña con hashcat sin utilizar el potfile, el comando será muy similar al anterior. La diferencia principal radicará en que se desactivará el potfile, y para lograrlo, utilizaremos el siguiente comando:

```
1 hashcat -m 22000 190198_1697598951.hc22000 rockyou_mod.dic --potfile-disable
```

Listing 12: Comando para obtener contraseña sin hashcat con potfile

Una vez que se ejecute el comando, se obtendrá el siguiente resultado:

```

tobias@tobias-NBKL-WA9X: ~/Documents/2023.2/Cryptography/Lab3$ hashcat -m 22000 190198_1697598951.hc22000 rockyou_mod.dic --potfile-disable
hashcat (v6.2.5) starting

OpenCL API (OpenCL 2.0 pocl 1.8 Linux, None+Asserts, RELOC, LLVM 11.1.0, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
* Device #1: pthread-AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx, 2539/5143 MB (1024 MB allocatable), 8MCU

Minimum password length supported by kernel: 8
Maximum password length supported by kernel: 63

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Slow-Hash-SIMD-LOOP

Watchdog: Temperature abort trigger set to 90c
Host memory required for this attack: 2 MB

Dictionary cache hit:
* Filename..: rockyou_mod.dic
* Passwords.: 11059707
* Bytes.....: 120106275
* Keyspace...: 11059707

1013acb976741b446d43369fb96dbf98:b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PWKID+EAPOL)
Hash.Target.....: 190198_1697598951.hc22000
Time.Started....: Wed Oct 18 00:56:12 2023 (1 sec)
Time.Estimated...: Wed Oct 18 00:56:13 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_mod.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 3870 H/s (0.09ms) @ Accel:64 Loops:256 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 2907/11059707 (0.03%)
Rejected.....: 1371/2907 (47.16%)
Restore.Point....: 1905/11059707 (0.02%)
Restore.Sub.#1...: Salt=0 Amplifier=0-1 Iteration=0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Magandaak00 -> Dangerous0
Hardware.Mon.#1...: Temp: 67c Util: 45%

Started: Wed Oct 18 00:56:11 2023
Stopped: Wed Oct 18 00:56:14 2023

```

Figura 12: Hashcat sin potfile

De la figura anterior, se puede observar la contraseña obtenida, la cual es “Security0”.

4.4. Identifica nomenclatura del output

A continuación se analiza cada campo del output:

- **Device:** Indica el dispositivo que se esta utilizando para realizar el ataque.
- **Password Lenght:** Indica el largo mínimo y máximo del formato compatible de las contraseñas que recibe hash.
- **Hashes:** Indica el número de hashes, de hashes unicos y de los salts únicos que se utilizan.
- **Rules:** Indica la cantidad de reglas utilizadas para el ataque.
- **Password length:** Indica la cantidad de optimizadores que se ocuparon el ataque.
- **Watchdog:** Indica el límite de temperatura, si la temperatura del dispositivo es de 90 grados Celcius el ataque se detendrá.
- **Host memory required:** Indica la cantidad de memoria necesaria para el ataque.
- **Dictionary cache built:** Indica detalles acerca del diccionario utilizado, como cantidad de contraseñas y el tamaño en bytes. Pero a diferencia del caso anterior, en este caso se hace uso de la memoria cache, ya que no existe el parámetro runtime, en cambio cuando se uso potfile este si existía y correspondía a 1 sec.
- **Hash descifrado:** Indica el hash descifrado
- **Session:** Indica información general de la sesión
- **Speed:** Velocidad con la que se descifra la información relacionada al rendimiento del ataque.
- **Started:** Inicio de ataque
- **Stopped:** Fin de ataque

No existen muchas diferencias en los campos, a excepción de lo que ya se mencionó. La otra diferencia importante es que no se crea el archivo “potfile.txt” con la contraseña obtenida.

4.5. Obtiene contraseña con aircrack-ng

Para obtener la contraseña con aircrack-ng, se empleará el siguiente comando. A diferencia de hashcat, la captura de handshake se utilizará en su formato original, es decir, en .pcap:

```
1 sudo aircrack-ng -a2 -w rockyou_mod.dic handshake.pcap
```

Listing 13: Comando para obtener contraseña con aircrack

4.6 Identifica y modifica parámetros solicitados por pycrack 4 DESARROLLO (PASO 3)

Una vez ejecutado se obtiene lo siguiente:

```
tobias@tobias-NBLK-WAX9:~/Documentos/2023.2/Cryptography/Lab3$ sudo aircrack-ng -a2 -w rockyou_mod.dic handshake.pcap
[sudo] contraseña para tobias:
Reading packets, please wait...
Opening handshake.pcap
Read 13 packets.

# BSSID      ESSID      Encryption
1  B8:48:7A:D2:DC:18  VTR-1645213  WPA (1 handshake)

Choosing first network as target.
Reading packets, please wait...
Opening handshake.pcap
Read 13 packets.
1 potential targets

Aircrack-ng 1.6

[00:00:01] 3401/9285363 keys tested (4304.38 k/s)
Time left: 35 minutes, 56 seconds      0.04%

KEY FOUND! [ Security0 ]

File names:
Master Key      : 55 E1 E0 F0 8E D7 53 80 F6 27 C6 DC 48 20 74 54
                  B7 54 98 37 71 FF C8 03 1D 89 C5 19 80 6F AC 76
Transient Key   : 3C 1B 89 A6 31 30 BA 04 B6 59 D9 7E 65 BD D2 07
                  9E C6 8D 2A D6 EF 7F 9E A1 95 1C BC CC 62 A6 5D
                  CC 07 B2 E3 9D 12 99 A7 66 D4 3C D7 61 56 53 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
EAPOL HMAC     : 18 13 AC B9 76 74 1B 44 6D 43 36 9F B9 6D BF 90
```

Figura 13: Contraseña obtenida con aircrack

Desde aquí, se puede observar que nuevamente se obtiene la contraseña, que es “Security0”, y esto sucede después de 3,401 intentos de contraseñas.

4.6. Identifica y modifica parámetros solicitados por pycrack

Para obtener la contraseña, primero es necesario instalar PyCrack utilizando el siguiente comando:

```
1 git clone https://github.com/nogilnick/PyCrack
```

Listing 14: Comando para instalar PyCrack

Una vez instalado, se abrirá el archivo “handshake.pcap” para su análisis. En este archivo, se podrán identificar los parámetros necesarios para utilizar en PyCrack, es decir, los campos que se modificarán. Los campos mencionados serán, SSID, aNonce, sNonce, apMac y cliMac, también serán manipulados los valores de mic y data de los paquetes específicos, todos estos campos son modificados en el script pywd.py.

4.6 Identifica y modifica parámetros solicitados por pycrack 4 DESARROLLO (PASO 3)

Nº	TIME	FROM	TO	PROTOCOL	LENGTH	INFO
1	0.000000	ee:de:67:8c:df:8b	ee:de:67:8c:df:8b	802.11	123	Association Request, SN=2292, FN=0, Flags=..... SSID=VIR-1045213
2	0.000002		ee:de:67:8c:df:8b	802.11	10	Acknowledgement, Flags=.....
3	0.002401	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	802.11	102	Association Response, SN=1184, FN=0, Flags=.....
4	0.002402		Tp-LinkT_d2:dc:18	802.11	10	Acknowledgement, Flags=.....
5	0.007381	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	133	Key (Message 1 of 4)
6	0.009336		Tp-LinkT_d2:dc:18	802.11	10	Acknowledgement, Flags=.....
7	0.017080	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	155	Key (Message 2 of 4)
8	0.017082		ee:de:67:8c:df:8b	802.11	10	Acknowledgement, Flags=.....
9	0.017087		ee:de:67:8c:df:8b	802.11	10	Clear-to-send, Flags=.....
10	0.050774	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	189	Key (Message 3 of 4)
11	0.050776		Tp-LinkT_d2:dc:18	802.11	10	Acknowledgement, Flags=.....
12	0.054559	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	133	Key (Message 4 of 4)
13	0.054560		ee:de:67:8c:df:8b	802.11	10	Acknowledgement, Flags=.....

Frame 1: 123 bytes on wire (984 bits), 123 bytes captured (984 bits)	0000 00 00 3a 01 b9 48 7a d2 dc 18 ee de 6
IEEE 802.11 Association Request, Flags=.....	0010 b9 48 7a d2 dc 18 40 8f 31 04 01 00 0
IEEE 802.11 Wireless Management	0020 52 2d 21 36 34 35 32 31 33 01 00 02 6
Fixed parameters (4 bytes)	0030 12 18 24 30 14 01 00 00 0f ac 04 01 6
Capabilities Information: 0x0431	0040 04 01 00 00 0f ac 02 00 00 32 04 30 4
Listen Interval: 0x0001	0050 10 51 51 53 54 73 74 75 76 77 78 7c 7
Tagged parameters (95 bytes)	0060 82 7f 05 04 00 00 01 dd 07 00 50 50 f
Tag: SSID parameter set: "VIR-1045213"	0070 00 dd 00 8c fd f0 00 10 00 00 00 00
Tag: Supported Rates 1(0), 2(0), 5.5(0), 11(0), 6, 9, 12, 18, [Mbit/sec]	
Tag: RSN Information	
Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]	
Tag: Supported Operating Classes	
Tag: Extended Capabilities (5 octets)	
Tag: Vendor Specific: Microsoft Corp.: WPA/WPAE: Information Element	
Tag: Vendor Specific: Qualcomm Inc.	

Figura 14: SSID

1 0.000000	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	802.11	123	Association Request, SN=2292, FN=0,
2 0.000002		ee:de:67:8c:df:8b	802.11	10	Acknowledgement, Flags=.....
3 0.002401	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	802.11	102	Association Response, SN=1184, FN=0,
4 0.002402		Tp-LinkT_d2:dc:18	802.11	10	Acknowledgement, Flags=.....
5 0.007381	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	133	Key (Message 1 of 4)
6 0.009336		Tp-LinkT_d2:dc:18	802.11	10	Acknowledgement, Flags=.....
7 0.017080	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	155	Key (Message 2 of 4)
8 0.017082		ee:de:67:8c:df:8b	802.11	10	Acknowledgement, Flags=.....
9 0.017087		ee:de:67:8c:df:8b	802.11	10	Clear-to-send, Flags=.....
10 0.050774	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	189	Key (Message 3 of 4)
11 0.050776		Tp-LinkT_d2:dc:18	802.11	10	Acknowledgement, Flags=.....
12 0.054559	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	133	Key (Message 4 of 4)
13 0.054560		ee:de:67:8c:df:8b	802.11	10	Acknowledgement, Flags=.....

Frame 5: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits)	0000 81
IEEE 802.11 QoS Data, Flags=.....F.	0010 b4
Logical-Link Control	0020 81
802.1X Authentication	0030 00
Version: 802.1X-2004 (2)	0040 5e
Type: Key (3)	0050 81
Length: 95	0060 00
Key Descriptor Type: EAPOL RSN Key (2)	0070 00
[Message number: 1]	0080 00
Key Information: 0x008a	
Key Length: 16	
Replay Counter: 1	
WPA Key Nonce: 4c2fb7eca28fba45accefd3ac5e433314270e04355b6d95086031b004a31935	
Key IV: 00000000000000000000000000000000	
WPA Key RSC: 00000000000000000000000000000000	
WPA Key ID: 00000000000000000000000000000000	
WPA Key MIC: 00000000000000000000000000000000	
WPA Key Data Length: 0	

Figura 15: aNonce

4.6 Identifica y modifica parámetros solicitados por pycrack 4 DESARROLLO (PASO 3)

1 0.000000	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	802.11	123 Association Request, SN=2292, FN=0,
2 0.000002		ee:de:67:8c:df:8b (- 802.11	10 Acknowledgement, Flags=.....	
3 0.002401	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	802.11	102 Association Response, SN=1184, FN=0,
4 0.002402		Tp-LinkT_d2:dc:18 (- 802.11	10 Acknowledgement, Flags=.....	
5 0.007381	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	133 Key (Message 1 of 4)
6 0.009336		Tp-LinkT_d2:dc:18 (- 802.11	10 Acknowledgement, Flags=.....	
7 0.017080	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	155 Key (Message 2 of 4)
8 0.017082		ee:de:67:8c:df:8b (- 802.11	10 Acknowledgement, Flags=.....	
9 0.017087		ee:de:67:8c:df:8b (- 802.11	10 Clear-to-send, Flags=.....	
10 0.050774	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	189 Key (Message 3 of 4)
11 0.050776		Tp-LinkT_d2:dc:18 (- 802.11	10 Acknowledgement, Flags=.....	
12 0.054559	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	133 Key (Message 4 of 4)
13 0.054560		ee:de:67:8c:df:8b (- 802.11	10 Acknowledgement, Flags=.....	

Frame 7: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits)		0000 81
IEEE 802.11 QoS Data, Flags:T		0010 bf
Logical-Link Control		0020 81
802.1X Authentication		0030 3f
Version: 802.1X-2001 (1)		0040 7f
Type: Key (3)		0050 00
Length: 117		0060 00
Key Descriptor Type: EAPOL RSN Key (2)		0070 00
[Message number: 2]		0080 00
Key Information: 0x010a		0090 00
Key Length: 0		
Replay Counter: 1		
WPA Key Nonce: 38bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c038f5869b96bbe9cdc		
Key IV: 00000000000000000000000000000000		
WPA Key RSC: 0000000000000000		
WPA Key ID: 0000000000000000		
WPA Key MIC: 1813acb976741b446d43369fb96dbf90		
WPA Key Data Length: 22		
WPA Key Data: 30140100000fac040100000fac040100000fac020000		

Figura 16: sNonce

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	802.11	123	Association Request, SN=2292,
2	0.000002		ee:de:67:8c:df:8b (- 802.11		10	Acknowledgement, Flags=.....
3	0.002401	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	802.11	102	Association Response, SN=1184
4	0.002402		Tp-LinkT_d2:dc:18 (- 802.11		10	Acknowledgement, Flags=.....
5	0.007381	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	133	Key (Message 1 of 4)
6	0.009336		Tp-LinkT_d2:dc:18 (- 802.11		10	Acknowledgement, Flags=.....
7	0.017080	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	155	Key (Message 2 of 4)
8	0.017082		ee:de:67:8c:df:8b (- 802.11		10	Acknowledgement, Flags=.....
9	0.017087		ee:de:67:8c:df:8b (- 802.11		10	Clear-to-send, Flags=.....
10	0.050774	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	189	Key (Message 3 of 4)
11	0.050776		Tp-LinkT_d2:dc:18 (- 802.11		10	Acknowledgement, Flags=.....
12	0.054559	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	133	Key (Message 4 of 4)
13	0.054560		ee:de:67:8c:df:8b (- 802.11		10	Acknowledgement, Flags=.....

[Protocols in frame: wlan]

IEEE 802.11 Association Request, Flags:

Type/Subtype: Association Request (0x0000)

Frame Control Field: 0x0000

.0000 0001 0011 1010 = Duration: 314 microseconds

Receiver address: Tp-LinkT_d2:dc:18 (b0:48:7a:d2:dc:18)

Destination address: Tp-LinkT_d2:dc:18 (b0:48:7a:d2:dc:18)

Transmitter address: ee:de:67:8c:df:8b (ee:de:67:8c:df:8b)

Source address: ee:de:67:8c:df:8b (ee:de:67:8c:df:8b)

BSS Id: Tp-LinkT_d2:dc:18 (b0:48:7a:d2:dc:18)

.... 0000 = Fragment number: 0

1000 1111 0100 = Sequence number: 2292

IEEE 802.11 Wireless Management

Fixed parameters (4 bytes)

Figura 17: apMac y cliMac

4.6 Identifica y modifica parámetros solicitados por pycrack 4 DESARROLLO (PASO 3)

7 0.017080	ee:de:67:8c:df:8b	Tp-Link_d2:dc:18	EAPOL	155 Key (Message 2 of 4)
8 0.017082		ee:de:67:8c:df:8b	(- 802.11	10 Acknowledgement, Flags=.....
9 0.017087		ee:de:67:8c:df:8b	(- 802.11	10 Clear-to-send, Flags=.....
10 0.050774	Tp-Link_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	189 Key (Message 3 of 4)
11 0.050776		Tp-Link_d2:dc:18	(- 802.11	10 Acknowledgement, Flags=.....
12 0.054559	ee:de:67:8c:df:8b	Tp-Link_d2:dc:18	EAPOL	133 Key (Message 4 of 4)
13 0.054560		ee:de:67:8c:df:8b	(- 802.11	10 Acknowledgement, Flags=.....

> Frame 7: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits) > IEEE 802.11 QoS Data, Flags:T > Logical-Link Control > 802.1X Authentication Version: 802.1X-2001 (1) Type: Key (3) Length: 117 Key Descriptor Type: EAPOL RSN Key (2) [Message number: 2] > Key Information: 0x010a Key Length: 0 Replay Counter: 1 WPA Key Nonce: 30bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c038f5869b96bbe9cdc Key IV: 00000000000000000000000000000000 WPA Key RSC: 0000000000000000 WPA Key ID: 0000000000000000 WPA Key MIC: 1813acb976741b446d43309fb96dbf90 WPA Key Data Length: 22 WPA Key Data: 3014010000fac040100000fac040100000fac020000	000 001 002 003 004 005 006 007 008 009
---	--

Figura 18: Primer Mic

7 0.017080	ee:de:67:8c:df:8b	Tp-Link_d2:dc:18	EAPOL	155 Key (Message 2 of 4)
8 0.017082		ee:de:67:8c:df:8b	(- 802.11	10 Acknowledgement, Flags=.....
9 0.017087		ee:de:67:8c:df:8b	(- 802.11	10 Clear-to-send, Flags=.....
10 0.050774	Tp-Link_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	189 Key (Message 3 of 4)
11 0.050776		Tp-Link_d2:dc:18	(- 802.11	10 Acknowledgement, Flags=.....
12 0.054559	ee:de:67:8c:df:8b	Tp-Link_d2:dc:18	EAPOL	133 Key (Message 4 of 4)
13 0.054560		ee:de:67:8c:df:8b	(- 802.11	10 Acknowledgement, Flags=.....

> Frame 10: 189 bytes on wire (1512 bits), 189 bytes captured (1512 bits) > IEEE 802.11 QoS Data, Flags:F. > Logical-Link Control > 802.1X Authentication Version: 802.1X-2004 (2) Type: Key (3) Length: 151 Key Descriptor Type: EAPOL RSN Key (2) [Message number: 3] > Key Information: 0x13ca Key Length: 16 Replay Counter: 2 WPA Key Nonce: 4c2fb7eca28fba45accefde3ac5e433314270e04355b6d95080031b004a31935 Key IV: 00000000000000000000000000000000 WPA Key RSC: cd00000000000000 WPA Key ID: 0000000000000000 WPA Key MIC: a349d01089960aa9f94b5857b0ea10c0 WPA Key Data Length: 58 WPA Key Data: db0eb43c3faf2c0e8b7e8a471f962c307e707e4718be724459167a88fa281f4d7ce38f01..	000 001 002 003 004 005 006 007 008 009 00a 00b
---	--

Figura 19: Segundo Mic

4.6 Identifica y modifica parámetros solicitados por pycrack 4 DESARROLLO (PASO 3)

7	0.017000	ee:de:67:8c:df:8b	Tp-Link-T_d2:dc:18	EAPOL	155 Key (Message 2 of 4)
8	0.017082		ee:de:67:8c:df:8b (- 802.11)		10 Acknowledgement, Flags=.....
9	0.017087		ee:de:67:8c:df:8b (- 802.11)		10 Clear-to-send, Flags=.....
10	0.050774	Tp-Link-T_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	189 Key (Message 3 of 4)
11	0.050776		Tp-Link-T_d2:dc:18 (- 802.11)		10 Acknowledgement, Flags=.....
12	0.054559	ee:de:67:8c:df:8b	Tp-Link-T_d2:dc:18	EAPOL	133 Key (Message 4 of 4)
13	0.054560		ee:de:67:8c:df:8b (- 802.11)		10 Acknowledgement, Flags=.....

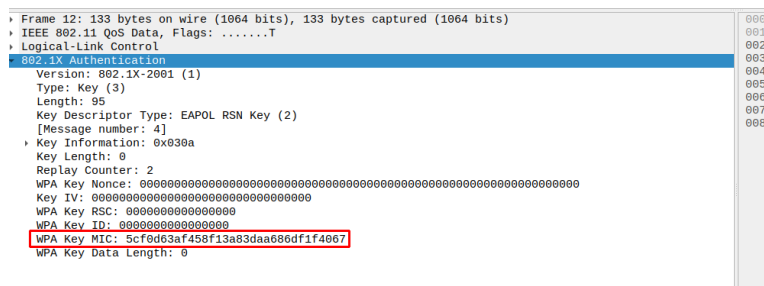


Figura 20: Tercer Mic

La data necesaria está conformada por tres paquetes, esta debe ser copiada en formato hexadecimal, a continuación se aprecia una de estas.

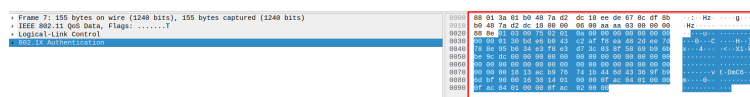


Figura 21: Data necesaria

Ahora que se conoce toda la información necesaria, se procede a modificar los campos con sus respectivos nuevos valores.

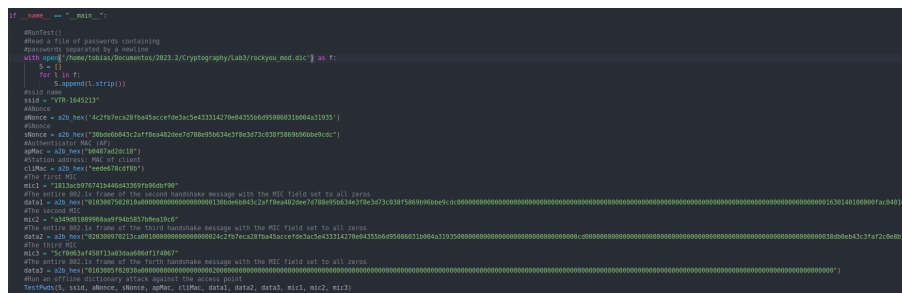


Figura 22: Script pywd.py modificado

Es importante mencionar que al momento de copiar la data esta es modificada justo en la extensión del MIC, ya que en esos 32 hexadecimales se reemplazará esa información con ceros.

4.7. Obtiene contraseña con PyCrack

Una vez modificado todo se compila el script por terminal y este retorna lo siguiente:

```
tobias@tobias-NBLK-WAX9X:~/Documentos/2023.2/Cryptography/Lab3/PyCrack$ python3 pywd.py
!!!Password Found!!!
Desired MIC1:      1813acb976741b446d43369fb96dbf90
Computed MIC1:     1813acb976741b446d43369fb96dbf90
Desired MIC2:      a349d01089960aa9f94b5857b0ea10c6
Computed MIC2:     a349d01089960aa9f94b5857b0ea10c6
Desired MIC2:      5cf0d63af458f13a83daa686df1f4067
Computed MIC2:     5cf0d63af458f13a83daa686df1f4067
Password: Security0
```

Figura 23: Contraseña obtenida con PyCrack

Desde aquí, se puede observar que nuevamente se obtiene la contraseña, que es “Security0”.

Conclusiones y comentarios

En este laboratorio, se exploraron técnicas de seguridad informática relacionadas con la identificación de vulnerabilidades en redes inalámbricas y la manipulación de contraseñas. Se puso de manifiesto la debilidad del cifrado WEP y se demostró su vulnerabilidad a través de herramientas como Aircrack-ng, Hashcat y el script pycrack. Estas herramientas proporcionaron una visión clara de por qué el cifrado WEP no se considera seguro en la actualidad, ya que su falta de robustez lo hace susceptible a ataques. Además, se adquirió experiencia en la modificación de diccionarios de contraseñas, una habilidad esencial en pruebas de seguridad.

El laboratorio subraya la importancia de adoptar cifrados más sólidos, como WPA2 o WPA3, para garantizar la protección de la privacidad y la confidencialidad de los datos en redes inalámbricas. Además, se adquirió conocimiento sobre herramientas que son fundamentales para identificar vulnerabilidades y comprender la fragilidad de sistemas de seguridad obsoletos.